



Resumen

Transcripción

[00:00] Y ahora una pregunta para ponernos a pensar con un poco más de detalle cómo es que se da esta ejecución. A ver, sabemos que la única puerta de entrada para registrar el salario de los funcionarios y calcular la bonificación total de todos es esta de aquí, que recibe como parámetro, ¿a quién? Al funcionario. Listo.

[00:28] Entonces, basado en esto, mi pregunta es: si yo estoy obteniendo la bonificación aquí, si me interesa aquí registrar el salario y obtener la bonificación del gerente y la bonificación del contador, en este momento, voy a buscar la clase, la perdí. Está aquí. ¿Cuál método se está ejecutando ahí?

[01:08] ¿El método `getBonificacion` de la clase `Funcionario`, que retorna el salario, el 5% del salario, o se está ejecutando el método `getBonificacion` de la clase que yo estoy definiendo? Vamos a hacerlo más interesante. Vamos a decirle aquí que él sobrescriba el método `getBonificacion` del contador. El contador va a tener una bonificación diferente.

[01:40] Para esto, yo puedo decir simplemente un `getB`, "Ctrl + espacio" y él aquí automáticamente ya me va a detectar el `getBonificacion`. ¿Por qué? Porque como extendiendo de `funcionario`, él aquí me va a preguntar. "Oye, ¿tú quieres sobrescribir el `getBonificacion` de casualidad? Y yo le voy a decir sí, quiero sobrescribir el `getBonificacion`."

[02:03] Y genero este código de aquí un poco complicado de entender, no se preocupen, esto lo vamos a ver con mucho más detalle en la siguiente clase.

quiero ahora centrar en un solo punto, y es este de aquí. Yo no voy a usar el método de la clase padre. Ya habíamos acordado que super era para decirle: "Oye, usa el método getBonificacion de la clase de arriba, en este caso funcionario". No.

[02:38] Yo quiero que aquí retorne simplemente 200. ¿Qué quiere decir? Que la bonificación del contador no va a ser porcentaje ni nada, va a ser una bonificación fija de 200. Perfecto. Si yo guardo y ejecuto mi test, vamos a ejecutar, y vemos que el cálculo ha cambiado. ¿Por qué? Porque de la última bonificación que él calculó aquí, 10600, ahora solamente le suma 200. Por lo cual nos da 10800. Hasta ahí todo bien.

[03:12] ¿Pero cuál es el método que él está llamando? ¿Qué pasa en ese caso, cuando tenemos el método, tanto en la clase padre como en la clase hija? ¿Qué sucede ahí? Sucede algo muy interesante y es esto de aquí. Él llama al método más específico. ¿Qué quiere decir esto?

[03:34] Si getBonificacion ya está en la clase Gerente, entonces él va a llamar al método de la clase Gerente, no va a llamar al método de la clase padre. Igual caso con el contador. Si él encuentra este método aquí, sobrescrito, entonces él va a llamar a este método de aquí, no va a llamar al método de la clase Funcionario. ¿Cómo se dieron cuenta que fue anteriormente que el contador tenía la misma bonificación del funcionario?

[04:05] Y como yo siempre digo, no basta solamente con crear. Vamos a probarlo. Entonces vamos a darle aquí un syso. Perfecto. Se fue. Ya. Y aquí le vamos a decir que nos indique nada más, ejecutando desde Gerente. Perfecto. "Ctrl + S" y la misma cosa aquí. A ver. Syso, comillas y ejecutando desde contador. Perfecto. Entonces, esto se va a imprimir si y solo si él está llamando estos métodos de las clases más específicas.

[05:01] Y vamos a probar nuestra teoría. Le damos ejecutar, salvamos aquí y vamos a ver qué nos tiene preparado esto. Y en efecto, esto fue ejecutado del

gerente, por eso obtuvo este valor, y esto fue ejecutado desde el contador, por eso obtiene el valor de 200. ¿Por qué? Nuevamente, a modo de repaso, porque la bonificación, el método `getBonificacion` si está implementado en la clase más específica, en este caso el gerente. ¿Por qué?

[05:32] Porque yo aquí declaré que Jimena era un gerente directamente, si la referencia, si esta referencia de aquí tiene el método, entonces él va a llamar a este método de aquí. Y si no lo tiene, lo que él va a hacer es subir un nivel más arriba y preguntarle directamente ¿a quién? Al funcionario, que es la clase padre, y es la que debería tener la respuesta. Entonces es así como funciona digamos la sobreescritura de los métodos y el polimorfismo.

[06:07] Esta es una aplicación clásica, clásica de polimorfismo. Vemos que esta clase con este método de registrar salario acepta cualquier objeto que sea del tipo funcionario. Esto lo van a ver muy seguido en el trabajo, entonces les recomiendo mucho practicarlo, si tienen alguna duda pueden postearlo en el foro y nos vemos en la siguiente clase.